Persist data outside of containers

In the previous step, you created a Dockerfile and executed a command. Now, you'll learn how to persist data outside of containers.

> **NOTE**
>
> This walk through teach you how to persist data outside of container. RunPod has the same concept used for attaching a Network Volume to your Pod.
>
> Consult the documentation on attaching a Network Volume to your Pod.

Why persist data outside of a container?

The key goal is to have data persist across multiple container runs and removals.

By default, containers are ephemeral - everything inside them disappears when they exit.

So running something like:

```
docker run busybox date > file.txt
```

Would only write the date to `file.txt` temporarily inside that container. As soon as the container shuts down, that file and data is destroyed. This isn't great when you're training data and want your information to persist past your LLM training.

Because of this, we need to persist data outside of the container. Let's take a look at a workflow you can use to persist data outside of a container.

Create a named volume

First, we'll create a named volume to represent the external storage:

```
docker volume create date-volume
```

Update Dockerfile

Next, we'll modify our Dockerfile to write the date output to a file rather than printing directly to stdout:

```
FROM busybox
WORKDIR /data
RUN touch current_date.txt
COPY entrypoint.sh /
RUN chmod +x /entrypoint.sh
ENTRYPOINT ["/entrypoint.sh"]
```

This sets the working directory to `/data`, touches a file called `current_date.txt`, and copies our script.

Update entrypoint script

The `entrypoint.sh` script is updated:

```
#!/bin/sh
date > /data/current_date.txt
```

This will write the date to the `/data/current_date.txt` file instead of printing it.

Mount the volume

Now when the container runs, this will write the date to the `/data/current_date.txt` file instead of printing it.

Finally, we can mount the named volume to this data directory:

```
docker run -v date-volume:/data my-image
```

This runs a container from my-image and mounts the `date-volume` Docker volume to the /data directory in the container. Anything written to `/data` inside the container will now be written to the `date-volume` on the host instead of the container's ephemeral filesystem. This allows the data to persist. Once the container exits, the date output file is safely stored on the host volume.

After the container exits, we can exec into another container sharing the volume to see the persisted data file:

```
docker run --rm -v date-volume:/data busybox cat /data/current_date.txt
```

This runs a new busybox container and also mounts the `date-volume`.

- Using the same `-v date-volume:/data mount` point maps the external volume dir to `/data` again.
- This allows the new container to access the persistent date file that the first container wrote.
- The `cat /data/current_date.txt` command prints out the file with the date output from the first container.
- The `--rm` flag removes the container after running so we don't accumulate stopped containers.

> **NOTE**
> Remember, this is a general tutorial on Docker. These concepts will help give you a better understanding on working with RunPod.

Edit this page

Previous
« Dockerfile

Next
Docker commands »

AI APIs

Community

Discord

Contact us

More

Blog

GitHub

Ask AI